

---

# **BACHELORARBEIT**

---

Herr  
**Jens Wittig**

**Konzeption und Realisierung  
einer TYPO3-Projektvorlage  
mit Responsive Design User  
Interface**

Mittweida, 2014



# **BACHELORARBEIT**

---

## **Konzeption und Realisierung einer TYPO3-Projektvorlage mit Responsive Design User Interface**

Autor:

**Herr**

**Jens Wittig**

Studiengang:

**Multimediatechnik**

Seminargruppe:

**MK08w1-B**

Erstprüfer:

**Prof. Dr.-Ing. Frank Zimmer**

Zweitprüfer:

**M.A. Thomas Richter**

Einreichung:

**Mittweida, 30.04.2014**

Verteidigung/Bewertung:

**Mittweida, 2014**

# **BACHELOR THESIS**

---

## **conception and realisation of a TYPO3-project template with responsive design user inter- face**

author:

**Jens Wittig**

course of studies:

**Multimedia Technology**

seminar group:

**MK08w1-B**

first examiner:

**Prof. Dr.-Ing. Frank Zimmer**

second examiner:

**M.A. Thomas Richter**

submission:

**Mittweida, 31.03.2014**

defence/evaluation:

**Mittweida, 2014**

## **Bibliografische Beschreibung:**

Wittig, Jens:

Konzeption und Realisierung einer TYPO3-Projektvorlage mit Responsive Design User Interface. - 2014.

Mittweida, Hochschule Mittweida, Fakultät Elektro- und Informationstechnik, Bachelorarbeit, 2014

## **Referat:**

Performance und Usability sind maßgebliche Anforderungen an die Konzeption von Webprojekten. Ständig erscheinen neue internetfähige Endgeräte auf dem Markt welche verschiedene Herangehensweisen an diese Anforderungen erfordern.

Um diesen gerecht zu werden, hat ein Umdenken in den Köpfen vieler (Web)Entwickler stattgefunden. Weg von pixelgenauen Darstellungen, hin zu flexiblen Webseiten, welche sich der verfügbaren Umgebung anpassen. Unter dem Begriff „Responsive Webdesign“ vereinen sich etablierten Techniken der Frontendentwicklung, sowie konzeptionelle Anpassungen.

Eine weitere Herausforderung ist es, Responsive Webdesign in bestehende Systeme zu integrieren. Diese Arbeit befasst sich mit verschiedenen Ansätzen der Nutzung responsiver Features in Zusammenhang mit dem Content Management System TYPO3.

Ergänzend entsteht ein responsiver Prototyp auf Basis einer bereits existierenden TYPO3-Grundkonfiguration.

# Inhalt

<b>Inhalt .....</b>	<b>I</b>
<b>Abbildungsverzeichnis .....</b>	<b>IV</b>
<b>Tabellenverzeichnis .....</b>	<b>VI</b>
<b>Abkürzungsverzeichnis .....</b>	<b>VII</b>
<b>1      Einleitung.....</b>	<b>1</b>
1.1 <i>Agenturbereich T-Systems MMS</i> .....	2
1.1.1      TYPO3 .....	2
1.1.2      Cleanpackage .....	3
1.2 <i>Ausgangslage</i> .....	4
1.3 <i>Zielstellung</i> .....	4
1.4 <i>Abgrenzung</i> .....	5
1.5 <i>Aufbau der Arbeit</i> .....	5
<b>2      Responsive Webdesign .....</b>	<b>6</b>
2.1 <i>Begriffserklärung</i> .....	6
2.1.1      Definition .....	6
2.1.2      technischer Hintergrund.....	6
2.1.3      Nutzen.....	7
2.2 <i>Fluides Layout</i> .....	7
2.2.1      Media Queries .....	7
2.2.2      „Responsive“ vs „Adaptive“ .....	8
2.3 <i>Flexible Schriftgrößen</i> .....	8
2.4 <i>Responsive Medien</i> .....	9
2.4.1      Responsive Bilder .....	9
2.4.1.1      Browsergerechte Anpassung.....	9
2.4.1.2      zusätzliches Nachladen.....	9
2.4.1.3      serverseitige Lösungen .....	10
2.4.2      Responsive Video .....	10
2.5 <i>Strategie</i> .....	10

2.5.1	Graceful Degradation / Desktop first .....	10
2.5.2	Progressive Enhancement / Mobile first .....	10
<b>3</b>	<b>Fachliche Konzeption .....</b>	<b>12</b>
3.1	<i>Layout</i> .....	12
3.1.1	Spalten .....	13
3.1.2	Navigation.....	13
3.2	<i>TYPO3 Inhaltselemente</i> .....	14
3.2.1	Typografie.....	14
3.2.1.1	Fließtext .....	14
3.2.1.2	Überschriften .....	15
3.2.2	Text mit Bild .....	15
3.2.2.1	im Text.....	16
3.2.2.2	in Spalte.....	17
3.2.2.3	mittig .....	18
3.2.3	News.....	19
3.2.4	Bilder .....	19
3.2.4.1	Slider .....	20
3.2.4.2	Galerie .....	21
3.2.4.3	Lightbox .....	21
3.2.3	Videos.....	22
3.2.4	Tabellen.....	24
3.2.3	Listen .....	24
3.2.3	Formulare .....	25
<b>4</b>	<b>Technische Konzeption.....</b>	<b>26</b>
4.1	<i>Frameworks</i> .....	26
4.2	<i>Layout</i> .....	27
4.2.1	Spalten .....	27
4.2.2	Navigation.....	27
4.3	<i>TYPO3 Inhaltselemente</i> .....	27
4.3.1	Typografie.....	27
4.3.2	Texte mit Bildern und News .....	28
4.3.3	Bilder .....	28
<b>5</b>	<b>Integration .....</b>	<b>29</b>
5.1	<i>Integration von Bootstrap</i> .....	29
5.1.1	Konfiguration und Auswahl der Komponenten .....	29
5.1.1.1	Media Queries breakpoints .....	30
5.1.1.2	Gridsystem.....	30
5.1.2	Implementierung in TYPO3.....	30

---

5.1.2.1	Einbinden der CSS-Datein .....	31
5.1.2.2	Einbinden der JavaScript-Datein .....	32
5.1.3	Templateanpassungen .....	32
5.2	<i>Integration magnificPopup</i> .....	32
5.3	<i>Integration responsive Images</i> .....	34
<b>6</b>	<b>Fazit</b> .....	<b>35</b>
	<b>Literatur und Quellenverzeichnis</b> .....	<b>36</b>
	<b>Selbstständigkeitserklärung</b>	



# Abbildungsverzeichnis

Abbildung 1-1: Prognose mobile Internetnutzung [Pmlmp2010] .....	1
Abbildung 1-2: Screenshot Cleanpackage .....	3
Abbildung 1-3: Screenshot der Seitenstruktur im Backend des Cleanpackage .....	4
Abbildung 2-1: Media Query Syntax .....	7
Abbildung 3-1: Main-, Service und Breadcrumbmenü bei Maximalansicht .....	13
Abbildung 3-2: expandiertes Mainmenü .....	14
Abbildung 3-3: Fließtext in Maximal-, Zwischen- und Minimalansicht .....	15
Abbildung 3-4: Text mit Bild links und rechts .....	16
Abbildung 3-5: Text mit Bild in Spalte links und rechts .....	17
Abbildung 3-6: Text mit Bild mittig .....	18
Abbildung 3-7: News-Vorschau und News-Detail .....	19
Abbildung 3-8: Slider in Maximal-, Zwischen- und Minimalansichtbbildung .....	20
Abbildung 3-9: Galerie in Maximal-, Zwischen- und Minimalansicht .....	21
Abbildung 3-10: Lightbox in Maximal-, und Minimalansicht .....	22
Abbildung 3-11: Video in Maximal-, und Minimalansichtl .....	23
Abbildung 3-12: Tabelle und Listen in Maximal-, und Minimalansicht .....	24
Abbildung 3 13: Formular in Maximal-, und Minimalansicht .....	25
Abbildung 5-1: Bootstrap jQuery Komponenten.....	29
Abbildung 5-2: Bootstrap LESS Komponenten.....	30

---

<b>Abbildung 5-3: Bootstrap HTML Navigationsstruktur .....</b>	<b>32</b>
<b>Abbildung 5-4: TYPO3 Image-Rendering .....</b>	<b>34</b>
<b>Abbildung 5 5: responsive Images HTML Struktur .....</b>	<b>34</b>

# Tabellenverzeichnis

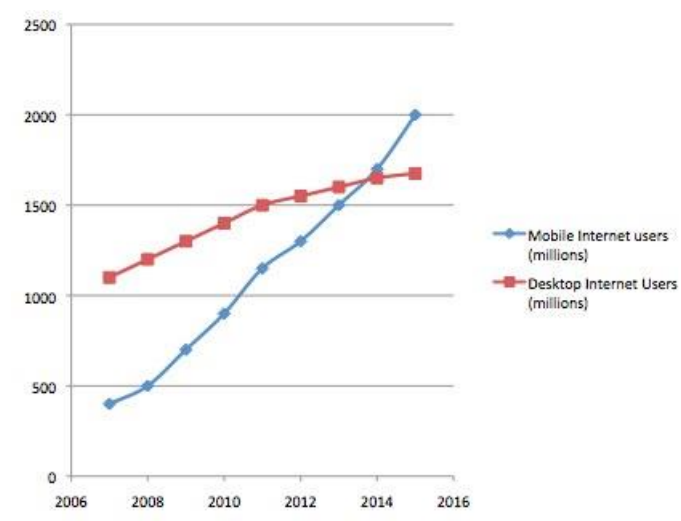
<b>Tabelle 3-1: Erläuterung zur Darstellungsform .....</b>	<b><a href="#">12</a></b>
<b>Tabelle 5-1: Bootstrap Verzeichnisse .....</b>	<b><a href="#">31</a></b>
<b>Tabelle 5-2: Bootstrap CSS Implementation .....</b>	<b><a href="#">31</a></b>
<b>Tabelle 5 3: Bootstrap JavaScript Implementation .....</b>	<b><a href="#">32</a></b>
<b>Tabelle 5-4: magnific Popup Verzeichnissebbildung.....</b>	<b><a href="#">33</a></b>
<b>Tabelle 5 5: magnific Popup CSS Implementation.....</b>	<b><a href="#">33</a></b>
<b>Tabelle 5 6: magnific Popup JavaScript Implementation .....</b>	<b><a href="#">33</a></b>
<b>Tabelle 5 7: responsive Images Verzeichniss .....</b>	<b><a href="#">34</a></b>

## Abkürzungsverzeichnis

<b>BP</b>	Break Point
<b>CMS</b>	Content Management System
<b>CSS</b>	Cascading Style Sheets
<b>HTML</b>	Hypertext Markup Language
<b>UHD</b>	Ultra High Definition
<b>PHP</b>	Hypertext Preprocessor

# 1 Einleitung

Die Nutzung des Internets hat sich in den letzten Jahren stark verändert. Neben den klassischen Desktop-Computern, sind es vor allem internetfähige Mobilgeräte, über welche immer mehr Zugriffszahlen verzeichnet werden.



**Abbildung 1-1: Prognose mobile Internetnutzung [Pmlmp2010]<sup>A1</sup>**

Diese Vielzahl an Geräten führt zu neuen Anforderungen an Websites. Wichtige Faktoren sind die User Experience<sup>1</sup>, die Usability<sup>2</sup> und die Performance<sup>3</sup>. Abhängig sind diese vor allem von der verfügbaren Displaygröße und den Browserkonfigurationen des verwendeten Gerätes. Bisher wurde für diese spezielle Form der Internetnutzung, parallel zur Desktopvariante, oft eine Mobile-Website<sup>4</sup>, bereitgestellt. Dies bedeutet meist erheblichen zu-

---

<sup>A1</sup> <https://www.3m5.de/nc/blog/beitrag/mobile-content-first/>

<sup>1</sup> Nutzungserlebnis der Anwender

<sup>2</sup> Gebrauchstauglichkeit

<sup>3</sup> Anzeigegeschwindigkeit

<sup>4</sup> Eigenständige Lösung getrennt von der eigentlichen Website. Siehe: <http://www.partnercommunication.de/webdesign-lexikon/mobile-webseite-app.php#.U2AQUFfJbh8>

sätzlichen Entwicklungsaufwand. Theoretisch müsste garantiert werden, dass die Informationen und Funktionen einer Webseite auf einem 2,5 Zoll Feature-Phone Display ebenso wie auf einem UHD-TV zur Verfügung stehen. Dazwischen gibt es weitere nahezu unzählige internetfähige Geräte wie Tablets, Netbooks, Smartphones, eBook-Reader und natürlich auch Desktop-Computer. Diese Geräte nutzen wiederum verschiedene Browser und Browserversionen. Es sollte zu erkennen sein dass der bisherige Standard, eine statische Webseite nach einem mit dem Kunden vereinbarten Designvorschlag zu entwickeln, diesen Anforderungen nicht gerecht werden kann.

Die Entwicklung muss hin zu flexiblen Webseiten führen. Inhalte müssen sich den Bildschirmdimensionen anpassen um eine übergreifende User Experience zu gewährleisten.

Diese Arbeit befasst sich mit der Analyse und der Integration flexibler Inhaltselemente anhand einer TYPO3 Projektvorlage des Agenturbereichs der T-Systems Multimedia Solutions Dresden.

## 1.1 Agenturbereich T-Systems MMS

Die T-Systems Multimedia Solutions (kurz MMS) ist hundertprozentige Tochter der T-Systems. Mit ca. 1500 Mitarbeitern an 7 Standorten handelt es sich um ein mittelständiges Unternehmen, welches in der Vergangenheit oft als Marktführer im Internet Agentur Ranking<sup>5</sup> ausgezeichnet wurde. Das Projektfeld 47: Creation&Production der MMS versteht sich als digitale Agentur für E-Business. Der Fokus liegt auf konzeptionellen Webentwicklungen aller Art. Konkret umfassen die Projekte Marketingkampagnen, Design und Entwicklung von Website-(Re-) Launches, sowie Spiele und Onlineshops. Beginnend bei der Konzeptionierung bis hin zu Roll-Outs, Entwicklung sowie Wartung und Pflege. Zum Einsatz kommt dabei häufig das Content Management System TYPO3<sup>6</sup>.

### 1.1.1 TYPO3

Wie bereits erwähnt, werden die meisten Projekte mit Hilfe des freien, auf PHP basierenden Content Management Systems TYPO3 umgesetzt. MySQL<sup>7</sup> Datenbanken werden

---

<sup>5</sup> <http://www.agenturranking.de/rankings/2012.html>

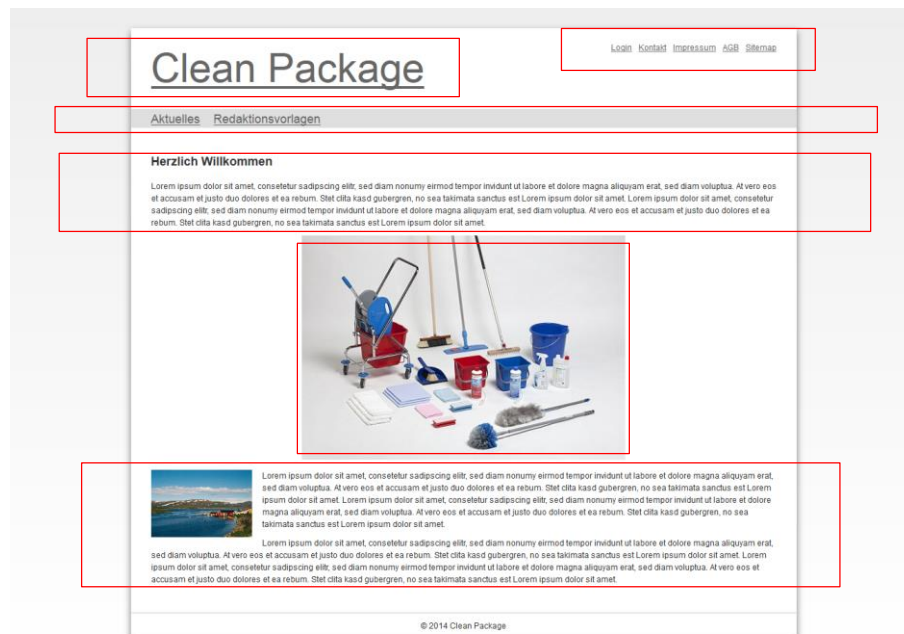
<sup>6</sup> Weitere Informationen zu TYPO3: siehe <http://typo3.org/>

<sup>7</sup> Weitere Informationen zur Open Source Datenbank MySQL: siehe <http://www.mysql.de/>

über das ebenfalls freie phpMyAdmin<sup>8</sup> verwaltet. Konfiguriert wird TYPO3 über ein Backend. Grundlegend hierbei ist die Konfigurationssprache TypoScript<sup>9</sup>. Funktional lässt sich das CMS durch Extensions<sup>10</sup> erweitern. Diese werden zum Teil selbst entwickelt oder über das sogenannte Repository<sup>11</sup> bezogen. Derzeit stehen mehr als 6000 freie Extensions zur Verfügung.

### 1.1.2 Cleanpackage

Das „Cleanpackage“ wurde von den Entwicklern der Abteilung erstellt und dient als Konfigurations-Vorlage für neue TYPO3-Projekte. Ziel ist es, die Basis-Setup-Aufwände gering zu halten und Learnings aus den Projekten kontinuierlich zu integrieren. Es umfasst neben der TYPO3 Grundkonfiguration die JavaScript Bibliotheken jQuery<sup>4</sup>, ColorBox<sup>5</sup>, Modernizr<sup>6</sup>, die Extensions Powermail<sup>7</sup> und Realurl<sup>8</sup> sowie ein simples CSS.



**Abbildung 1-2: Screenshot vom Frontend des Cleanpackage. Zu sehen sind die Startseite mit den statischen Elementen Überschrift, Service Navigation, Hauptnavigation und den TYPO3 Inhaltselementen „Text“, „Bild“, sowie „Text mit Bild links im Text“ (v.l.n.r.).**

<sup>8</sup> Weitere Informationen zu phpMyAdmin: siehe [www.phpmyadmin.net/](http://www.phpmyadmin.net/)

<sup>9</sup> Weitere Informationen zu TypoScript: siehe <http://www.typo3.net/tsref>

<sup>10</sup> Weitere Informationen zu Extensions (siehe <http://www.typo3-websites.eu/typo3-erweiterungen.htm>)

<sup>11</sup> TYPO3 Repository (siehe <http://typo3.org/extensions/repository>)

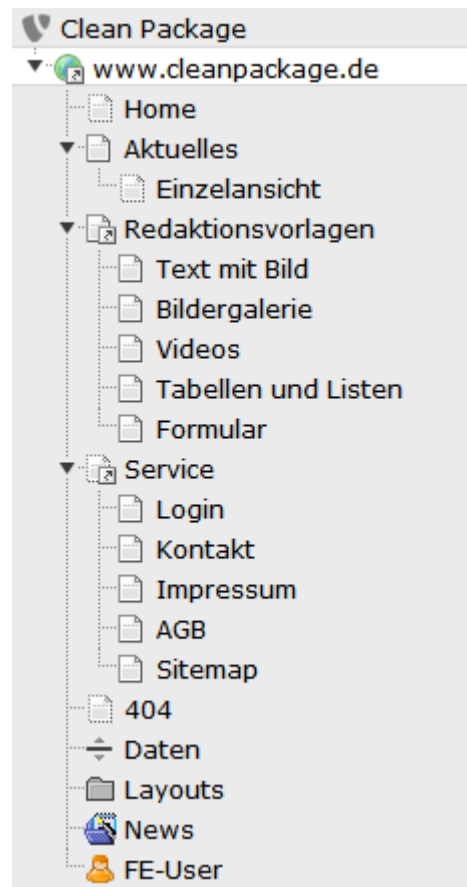


Abbildung 1-3: Screenshot der Seitenstruktur im Backend des Cleanpackage

## 1.2 Ausgangslage

Das Thema Responsive Design wurde in der Vergangenheit oft innerhalb der Abteilung thematisiert. Die Komplexität von TYPO3 sowie die Positionierung des Themas bei Kunden waren bisher stets Gründe, warum Responsive Design in den Projekten noch nicht eingesetzt wurde. Mittlerweile wird Responsive Design vom Kunden gefordert. Das Thema nimmt momentan einen hohen Stellenwert im Projektalltag ein, ohne dass eine technische Richtlinie existiert. Hier soll diese Bachelorthesis anknüpfen.

## 1.3 Zielstellung

Ziel ist es ein Responsive Design User Interface<sup>1</sup> für das Cleanpackage bereit zu stellen, welches bestenfalls als Grundlage für zukünftige TYPO3 Projekte mit dieser Funktionalität dienen soll. Diese umfasst die dynamische Darstellung der TYPO3 Inhaltselemente und

---

<sup>1</sup> Grafische Benutzeroberfläche



Optimierung der Performance durch adaptive Bildgenerierung<sup>2</sup>. Dafür werden verschiedene Ansätze analysiert und auf deren Basis die nötigen Anpassungen am Cleanpackage dokumentiert.

## 1.4 Abgrenzung

Der Fokus dieser Arbeit liegt zu großen Teilen auf der Integration der Funktionalitäten. Zu folgenden Punkten kann im zeitlichen Rahmen dieser Arbeit nur annähernd Stellung bezogen werden:

- Neben den technischen Aspekten sind für die Umsetzung von Responsive Design ebenso Anpassungen in den Projektprozessen nötig. Unter 2.5 wird auf die Strategien „Mobile first“ sowie „Desktop first“ eingegangen. Da das vorhandene Cleanpackage durch Anpassung erweitert wird, kann nur der bereits angewandte Desktop-first-Ansatz weiter geführt werden.
- Steigerung von Usability, User Experience und Performance werden in der Konzeption angesprochen und behandelt. Eine tiefgründige (Software-)Analyse und Optimierung hinsichtlich der umfangreichen Konfigurationsmöglichkeiten der verwendeten Frameworks, Extensions und JavaScript Bibliotheken kann im Rahmen dieser Bachelorarbeit nicht stattfinden. Weiterhin sind einige Konfigurationsparameter natürlich projektabhängig.
- Auf die in der fachlichen Konzeption zu responsiven Videos (3.2.5) angesprochene Optimierung der Videoauslieferung, nach dem Vorbild der adaptiven Bilder, muss auf Grund des hohen technischen Aufwandes verzichtet werden.
- Der Anspruch auf Aktualität und Verfügbarkeit der verwendeten Lösungen besteht nur zum Zeitpunkt der Abgabe dieser Abschlussarbeit.

## 1.5 Aufbau der Arbeit

Nach einer Einführung in das Thema Responsive Design widmet sich Kapitel 3 der fachlichen Konzeptionierung. Dafür werden die mit der Abteilung vereinbarten Anforderungen an konkreten Inhaltselementen (Bildern, Tabellen, Formularen, etc.) festgelegt. Zu diesen Anforderungen werden in Kapitel 4 technische Lösungswege aufgezeigt. Deren Implementierung ins Cleanpackage wird anschließend dokumentiert bevor im letzten Kapitel ein kritisches Resümee über die erreichten Ziele und ein Ausblick auf zukünftige Entwicklung erfolgt.

---

<sup>2</sup> Geräteoptimierte Auflösung

## 2 Responsive Webdesign

„There is no Mobile Web. There is only The Web, which we view in different ways. There is also no Desktop Web. Or Tablet Web.“ - Stephen Hay<sup>1</sup> auf Twitter.

### 2.1 Der Begriff Responsive Design

Ursprünglich geprägt hat den Begriff „Responsive Design“ der Bostoner Designer und Entwickler Ethan Marcotte in einem Artikel für das Webmagazin „A List Apart“ im Mai 2010<sup>2</sup>. Er beschreibt dort die Grundzüge der Technik und wirbt für ein neues Verständnis von Webdesign im Angesicht der Zunahme mobiler Geräte.

#### 2.1.1 Definition

Responsive Design umfasst verschiedene konzeptionelle und technische Aspekte welche auf die Optimierung von Webseiten auf verschiedenen Endgeräten abzielen. Dies betrifft vor allem die Flexibilität von Content bei unterschiedlichen Bildschirmgrößen, sowie - Varianten, und die Geschwindigkeit, mit welcher diese an den Client ausgeliefert werden.

Eine responsive Website hat laut Marcotte folgende drei Eigenschaften:

- fluides Layout
- flexible Inhalte
- Verwendung von Media Queries<sup>3</sup>

#### 2.1.2 Technischer Hintergrund

Im Grundlegenden basieren alle Lösungen zu Responsive Design auf HTML, CSS und JavaScript.

---

<sup>1</sup> Stephen Hay ist ein bekannter Frontendentwickler und Autor [StHay]

<sup>2</sup> <http://alistapart.com/article/responsive-web-design/> [EtMa]

<sup>3</sup> Ein Modul aus der CSS3 - Spezifikation

### 2.1.3 Nutzen

- Erhöhung der User Experience durch optimierte Darstellung des Contents auf verschiedenen Endgeräten
- Erhöhung der Usability durch endgeräteabhängige Navigation
- Optimierung von Ladezeiten
- Verringerung der Entwicklungskosten durch Verwendung bestehender Lösungen

## 2.2 Fluides Layout

Die Bezeichnung deutet bereits darauf hin, dass ein fluides Layout, im Gegensatz zu einem statischen, nicht starr in der Anordnung von Spalten und Inhalten verharret. Vielmehr passt sich das Layout dem vorhandenen Viewport<sup>4</sup> an und skaliert dabei Inhalte. Zurückzuführen ist dies auf die Verwendung von Weitenangaben in Prozentsätzen statt in Pixeln.

### 2.2.1 Media Queries

Anhand von Eigenschaften wie Breite, Ausrichtung und Auflösung eines Gerätes kann man mit Hilfe von Media Queries steuern, welche Stilregeln verwendet werden. Im Zusammenhang mit Responsive Design, kann beispielsweise festgelegt werden, ab welcher Breite sich das Layout einer Website von zweispaltig auf einspaltig ändern soll, um für ein mobiles Display oder ein verkleinertes Browserfenster übersichtlicher zu sein. Dafür legt man je nach Ansatz einen Punkt fest, an welchem eine neue Media Query mit angepassten Style-Regeln in Kraft tritt. Ein solcher Umschaltpunkt wird als Breakpoint (BP) bezeichnet.

Die allgemeine Syntax<sup>5</sup> einer Media Query lautet:

```
@media [not|only] Typ [and] (Ausdruck) { Regeln }
```

**Abbildung 2-1: Media Query Syntax**

---

<sup>4</sup> Verfügbarer Anzeigebereich

<sup>5</sup> <http://www.w3.org/TR/css3-mediaqueries/#syntax>

- Media: der gewünschte Gerätetyp. In der Praxis werden fast ausschließlich all, screen und print verwendet.
- Medienausdruck: Test auf ein Feature, welcher entweder true oder false ergibt.
- Logischer Ausdruck: Schlüsselwörter für komplexere Ausdrücke wie „and“, „or“, „not“, „only“.
- Regeln: CSS Regeln

### 2.2.2 „Responsive“ vs „Adaptive“

Ein weiterer Kunstbegriff im Zusammenhang mit flexiblen Webseiten ist das „Adaptive“ Webdesign. Eine adaptive Website ist gegenüber einer responsiven Seite nur bedingt flexibel. Es kommen ebenfalls Media Queries zum Einsatz, welche die Anordnung der Elemente zwischen festen Breakpoints definieren. Die Elemente selbst bleiben jedoch zwischen diesen Grenzen starr. Da die MediaQueries nicht für alle verfügbaren Endgeräte festgelegt werden können, ist dieser feine Unterschied der Nachteil gegenüber einem voll flexiblen responsiven Layout.

## 2.3 Flexible Schriftgrößen

Auch Schrift soll flexibel dargestellt werden. Es muss stets gewährleistet sein, dass der Text lesbar ist und gegebenenfalls durch Skalierung einer Ansicht seine Schriftgröße anpasst. Mittel der Wahl sind auch hier Schriftmaße in Prozent oder „em“<sup>6</sup>. Im Gegensatz zu Schriftgrößen in Pixeln sind diese flexibel skalierbar. Sie sind zudem kaskadierbar und bieten weitere Vorteile bei der Umsetzung von Barrierefreiheit sowie bei der Verwendung von Displays mit verschiedenen Pixeldichten<sup>7</sup>.

Moderne Einheiten wie „rem“, „vw“, „vh“, „vmin/vmx“<sup>8</sup> bieten weitere Vorteile, kommen momentan allerdings auf Grund der noch zu geringen Browserunterstützung<sup>9</sup> nicht für die Verwendung in einer Projektvorlage in Frage.

---

<sup>6</sup> Weitere Informationen: <http://de.selfhtml.org/css/formate/wertzuweisung.htm>

<sup>7</sup> Beispielsweise die von Apple verwendeten Retina-Displays

<sup>8</sup> Weitere Informationen: <http://css-tricks.com/the-lengths-of-css/>

<sup>9</sup> Weitere Informationen: <http://caniuse.com/viewport-units>

## 2.4 Responsive Medien

Großes Augenmerk liegt auf der Anpassung der verschiedenen medialen Inhalte wie Bilder und Videos. Diese haben aufgrund ihres meist hohen Datenvolumen einen signifikanten Einfluss auf Latenz einer Website. In der Regel besitzen sie feste Abmessungen sowie Auflösung und benötigen deshalb eine entsprechende Strategie um responsiv dargestellt zu werden.

### 2.4.1 Responsive Bilder

Es gibt mehrere Anforderungen welche Responsive Images erfüllen sollen. Ihre Darstellung soll anhand des verfügbaren Viewports flexibel erfolgen (BILD). Gleichzeitig soll ihre Datenmenge optimiert für das verwendete Gerät und die verfügbare Netzstärke sein. Dabei gibt es drei Stellschrauben, welche vor der Auslieferung entsprechend angepasst werden können:

- Abmessungen: Die Breite und Höhe des Bildes
- Pixeldichte: Hochauflösende Displays wie die in einigen Apple Geräten verwendeten Retina-Displays besitzen eine höhere Pixeldichte als klassische Displays. So bleiben die verwendeten Abmessungen des Bildes gleich, jedoch werden die Bilder schärfer dargestellt und benötigen mehr Speicher
- Datenvolumen: Menge an Daten welche übertragen werden müssen um das Bild darzustellen. Auch von den bereits genannten noch Faktoren abhängig

Im Allgemeinen gibt es folgende drei Herangehensweisen responsive Bilder umzusetzen<sup>1</sup>.

#### 2.4.1.1 Browsergerechte Anpassung

Die meisten Frontendlösungen versuchen bestmöglich festzulegen, welches Bild geladen werden soll, bevor der Browser ein ungeeignetes Bild lädt. Da Browser ihrerseits verschiedene Methoden<sup>2</sup> verwenden um eine Website möglichst schnell darzustellen und Bilder zu laden, besteht die Schwierigkeit darin, diese Maßnahmen möglichst zu umgehen.

#### 2.4.1.2 zusätzliches Nachladen

Bei dieser Strategie entscheidet man sich prinzipiell für Bilder welche für die kleinste Bildschirmvariante geeignet sind. Bei Bedarf wird zusätzlich ein größeres Bild

---

<sup>1</sup> Praxiswissen Responsive Webdesign S.112 [TiKa2013]

<sup>2</sup> Weitere Informationen: <http://www.golem.de/news/google-chrome-27-laedt-webseiten-schneller-1305-99372.html>

nachgeladen. Dies geschieht, aufgrund der doppelten Requests, zwar zu Lasten der Performance, jedoch hält sich der Konfigurationsaufwand in Grenzen.

#### **2.4.1.3 serverseitige Lösungen**

Durch verschiedene Methoden kann vor der Auslieferung an den Clienten durch ausgelesene Geräteinformationen ermittelt werden welches Bild geladen werden soll. Dabei handelt es sich jedoch um eine wenig zukunftsfreundliche Variante, da bei der Vielzahl von Geräten keine verlässlichen Informationen übermittelt werden.

### **2.4.2 Responsive Videos**

Prinzipiell gelten für Videos die gleichen Anforderungen wie für Bilder. Flexible Abmessungen sowie optimale Datenmenge.

## **2.5 Strategie**

Im Zusammenhang mit Responsive Design und steigender mobiler Internetnutzung, lohnt es sich, neben dem technischen Aspekt, auch die Strategie im Projektmanagement zu hinterfragen.

### **2.5.1 Graceful Degradation / Desktop first**

Dieser Ansatz entspricht der gewohnten Herangehensweise. Hierbei werden die Inhalte und unterstützten Funktionen meist für eine Desktop Website konzeptioniert. Für kleinere oder ältere Endgeräte werden durch die „würdevolle Verschlechterung“ Features, welche zuvor implementiert wurden, wieder deaktiviert oder umgebaut.

Im Projekt wird bei dieser Herangehensweise ein Designentwurf an die Entwickler abgegeben, welche diesen dann getreu der Vorlage umsetzen. Da diese Strategie meist hohen Aufwand bedeutet, sollte man gut überlegen, ob man diesen Ansatz beibehalten will.

### **2.5.2 Progressive Enhancement / Mobile first**

Progressive Enhancement bedeutet so viel wie „zunehmende Verbesserung“. Der sogenannte „Mobile first“-Ansatz ist im Prinzip die Umsetzung des Progressive Enhancement. Hier wird, im Gegensatz zum Graceful Degradation, zunächst der minimalste Entwicklungsaufwand betrieben<sup>3</sup>. Ziel ist es, das „schlechteste“ Zielmedium zu bedienen und die grundlegendsten Inhalte und Funktionen für alle Endgeräte zur

---

<sup>3</sup> Weitere Informationen; <http://accessites.org/site/2007/02/graceful-degradation-progressive-enhancement/>

Verfügung zu stellen. Ist dieser Schritt vollzogen wird nun für leistungsfähigere Geräte/Browser zusätzlicher Entwicklungsaufwand betrieben. Oft wird in dieser Phase erkannt das kaum noch funktionelle Anpassungen nötig sind. Meist bedarf es nur noch einer Optimierung der User Experience für leistungsfähigere Geräte.

Hierbei empfiehlt es sich agiles Projektmanagement<sup>4</sup> zu betreiben, bei welchem Designer und Entwickler zusammen an der Umsetzung arbeiten. Im Zusammenhang mit Responsive Design gibt es zwar erste Lösungen, wie Edge Reflow<sup>5</sup> der Firma Adobe, um responsive Designvorlagen zu erstellen, jedoch sind diese für den Projektalltag noch nicht ausgereift.

Die Vorteile von Mobile first sind in der Regel die Reduzierung des Entwicklungsaufwands, verkürzte Projektlaufzeiten und Erhöhung der Usability.

---

<sup>4</sup> Weitere Informationen: <http://t3n.de/magazin/praxisbericht-scrum-kanban-scrumbuts-agiles-232822/>

<sup>5</sup> Weitere Informationen: <http://html.adobe.com/edge/reflow/>

### 3 Fachliche Konzeption

Im Folgenden Abschnitt wird festgelegt, welche Anpassungen am Cleanpackage vorgenommen werden sollen. Es wird aufgezeigt wie sich die TYPO3 Inhaltselemente, das Layout und die Navigation bei Veränderung des Viewports verhalten sollen.

Die zu den TYPO3 Inhaltselementen verwendeten Gerätegrafiken dienen zur sinngemäßen Darstellung der Neu-Anordnung, beziehungsweise des Verhaltens, der Inhaltselemente bei verschiedenen Displaygrößen

Endgerät	Dargestellte Ansicht	Einordnung Viewport/BP
Desktop Bildschirm	Maximalansicht	Viewport > höherer BP
IPad	Zwischenansicht	niedriger BP < Viewport > höherer BP
IPhone	Minimalansicht	Viewport < niedriger BP

**Tabelle 3-1: Erläuterung zur Darstellungsform**

Die Beschreibungen über die Anpassungen erfolgen analog dieser Festlegung von der weitesten zur schmalsten Darstellung. Queransichten mobiler Geräte hätten je nach Festlegung einen Wechsel zur nächst höheren Ansicht und/oder zur Skalierung der Inhaltselemente zur Folge und werden nicht gesondert aufgeführt.

#### 3.1 Layout

Grundsätzlich sollen zwei Breakpoints existieren. Diese werden Content abhängig<sup>1</sup> gesetzt, so dass der Fluss der Seite erhalten bleibt. Das Unterschreiten dieser Punkte soll zu Änderungen der Spaltenanzahl, der Navigationsstruktur, sowie Content Anordnung erfolgen.

---

<sup>1</sup> Weitere Informationen: <http://blog.kulturbanause.de/2013/12/content-choreography-fur-responsive-websites/>



### 3.1.1 Spalten

Je nach Seitenlayout existieren im Cleanpackage maximal zwei Spalten (Hauptcontent und Randspalte Rechts). Die horizontalen Grenzen der Hauptspalte agieren jederzeit flexibel. Wird der erste BP unterschritten, soll die rechte Spalte unterhalb der Hauptspalte dargestellt werden und sich ebenso flexibel verhalten.

### 3.1.2 Navigation

Es existieren drei Navigationselemente: Das Hauptmenü mit den Hauptmenü- und Submenüpunkten, das Servicemenü und das Breadcrubmenü. Bei voller Breite (oberhalb des höheren BP) werden die Hauptmenüpunkte horizontal nebeneinander dargestellt. Bei Auswahl bzw. Mouseover-Geste werden gegebenenfalls Untermenüpunkte in vertikaler Ausrichtung darunter angezeigt. Das Service Menü befindet sich oben rechts und wird ebenso wie das Breadcrumb Menü horizontal dargestellt. Letzteres befindet sich direkt unterhalb der Hauptnavigation. Wird der erste BP unterschritten, soll das Hauptmenü kompakter dargestellt werden. Die Untermenüpunkte sollen nun „ausklappbar“ sein (Pulldown-Menü<sup>2</sup>). Das Service Menü soll unter den Content floaten. Ebenso wie beim Breadcrubmenü, zu welchem keine Anpassungen erfolgen, sollen hier Umbrüche vermieden werden. Eine Betrachtung der Zwischenansicht ist hier nicht notwendig. Die Anordnung bleibt erhalten und die Elemente agieren flexibel in ihrer Weite und Schriftgröße.



**Abbildung 3-1: Main-, Service und Breadcrubmenü bei Maximalansicht**

<sup>2</sup> Weiterer Informationen: <http://www.itwissen.info/definition/lexikon/Pull-Down-Menue-pull-down-menue.html>



Abbildung 3-2: expandiertes Mainmenü und Breadcrumbmenü in der Minimalansicht

## 3.2 TYPO3 Inhaltselemente

Im Folgenden werden die Anforderungen der im Cleanpackage vorkommenden Standard-Inhaltselemente bei Viewportänderung vereinbart.

### 3.2.1 Typografie

Es soll stets eine gute Lesbarkeit des Textes, unabhängig des verwendeten Displays bzw. Browsers, gegeben sein.

#### 3.2.1.1 Fließtext

Textfelder sollen in ihren Abmessungen jederzeit voll flexibel agieren. Textzeilen sollen bei Änderungen der Fensterbreite selbstständig umbrechen.



Abbildung 3-3: Fließtext in Maximal-, Zwischen- und Minimalansicht

### 3.2.1.2 Überschriften

Überschriften sollen lediglich drei definierbare Schriftgrößen besitzen welche bei Änderung der Ansicht wechseln. Durch die Wahl der Breakpoints wird gewährleistet, dass die Überschriften jederzeit vollständig sichtbar sind und nicht durch Viewportgrenzen abgeschnitten werden.

### 3.2.2 Text mit Bild

Weitere Standard TYPO3-Inhaltselemente sind „Text mit Bild“. Unterschieden wird je nach Anordnung. Die jeweilige Struktur soll bis zum niedrigsten Breakpoint beibehalten werden. Dabei werden Text und Bild zwischen den Breakpoints durchgängig skaliert. Nach Unterschreitung des niedrigsten Breakpoints „rutscht“ das Bild unter den Text und alle Text-mit-Bild-Elemente werden gleich dargestellt.

### 3.2.2.1 im Text

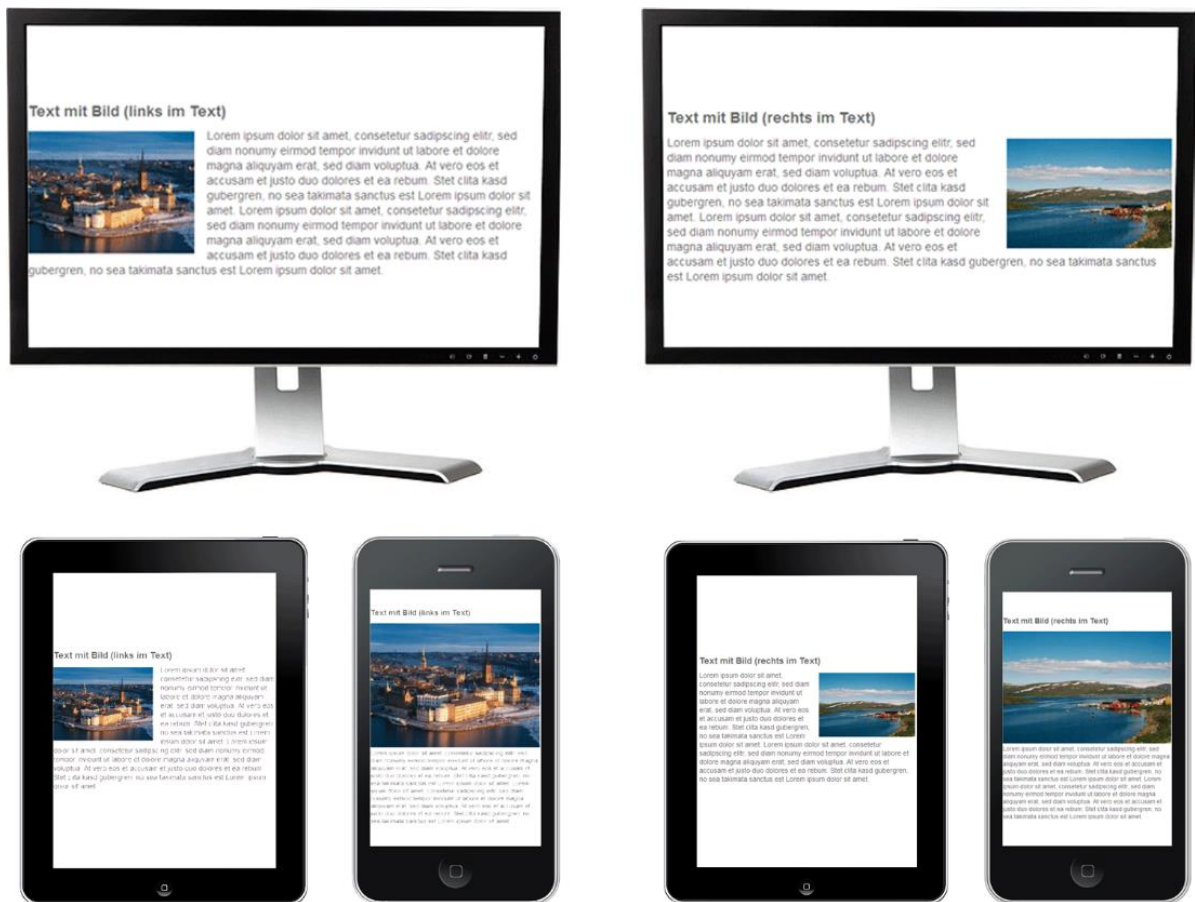
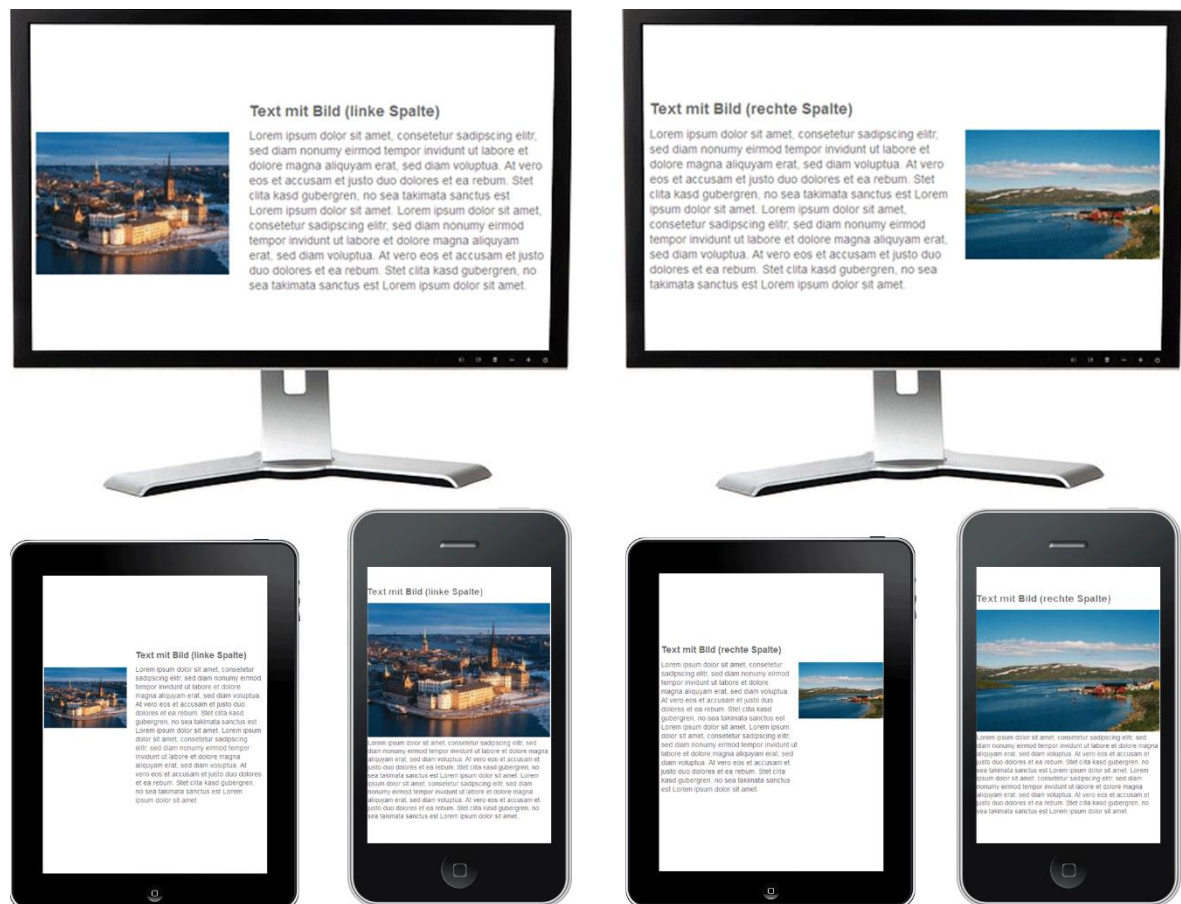


Abbildung 3-4: Text mit Bild links und rechts in Maximal-, Zwischen- und Minimalansicht

### 3.2.2.2 in Spalte



**Abbildung 3-5: Text mit Bild in Spalte links und rechts in Maximal-, Zwischen- und Minimalansicht**

### 3.2.2.3 mittig



Abbildung 3-6: Text mit Bild mittig in Maximal-, Zwischen- und Minimalansicht

### 3.2.3 News

Das Verhalten von News entspricht dem der Text-mit-Bild-links-Elemente. Man unterscheidet zwischen der Vorschau- und der Detailvariante.

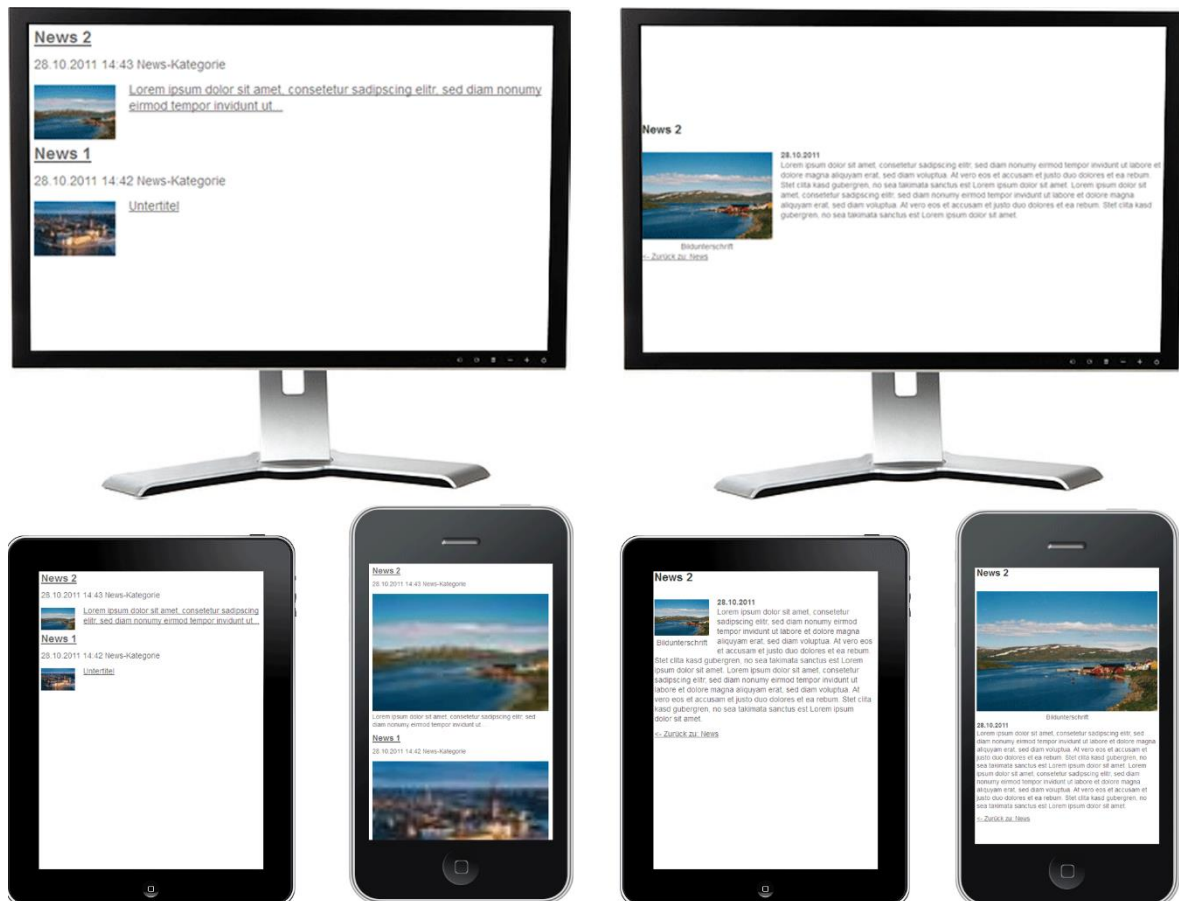


Abbildung 3-7: News-Vorschau und News-Detail in Maximal-, Zwischen- und Minimalansicht

### 3.2.4 Bilder

Bilder sollen zwischen den Breakpoints flexibel ihre Größe anpassen. Ein weiteres Kriterium ist die Nutzung adaptiver Bilder, welche optimaler Weise in Abhängigkeit vom verwendeten Display, der verfügbaren Internetverbindung und des verfügbaren Viewports geladen werden.

### 3.2.4.1 Slider



Abbildung 3-8: Slider in Maximal-, Zwischen- und Minimalansicht



### 3.2.4.2 Galerie

Mehrere Bilder werden je nach Anzahl und Viewport in variabler Spaltenanzahl (6-3-2) dargestellt. Zwischen den Breakpoints werden sie stets flexibel skaliert.

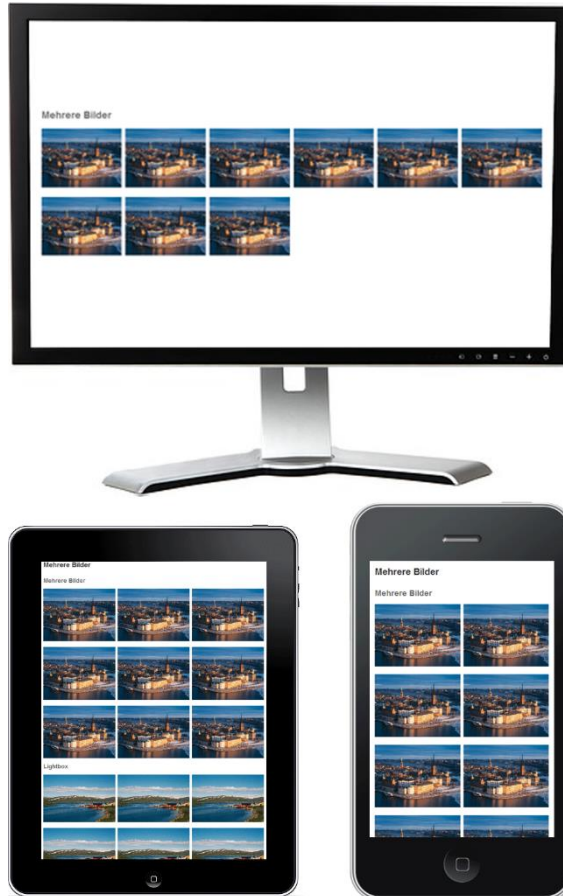


Abbildung 3-9: Galerie in Maximal-, Zwischen- und Minimalansicht

### 3.2.4.3 Lightbox

Wird ein Bild im Backend als Lightbox deklariert, so erfolgt nach Anklicken eine Vollbilddarstellung, welche sich entsprechend der Fensterbreite skaliert. Die Steuerungselemente sollen entsprechend positioniert werden.



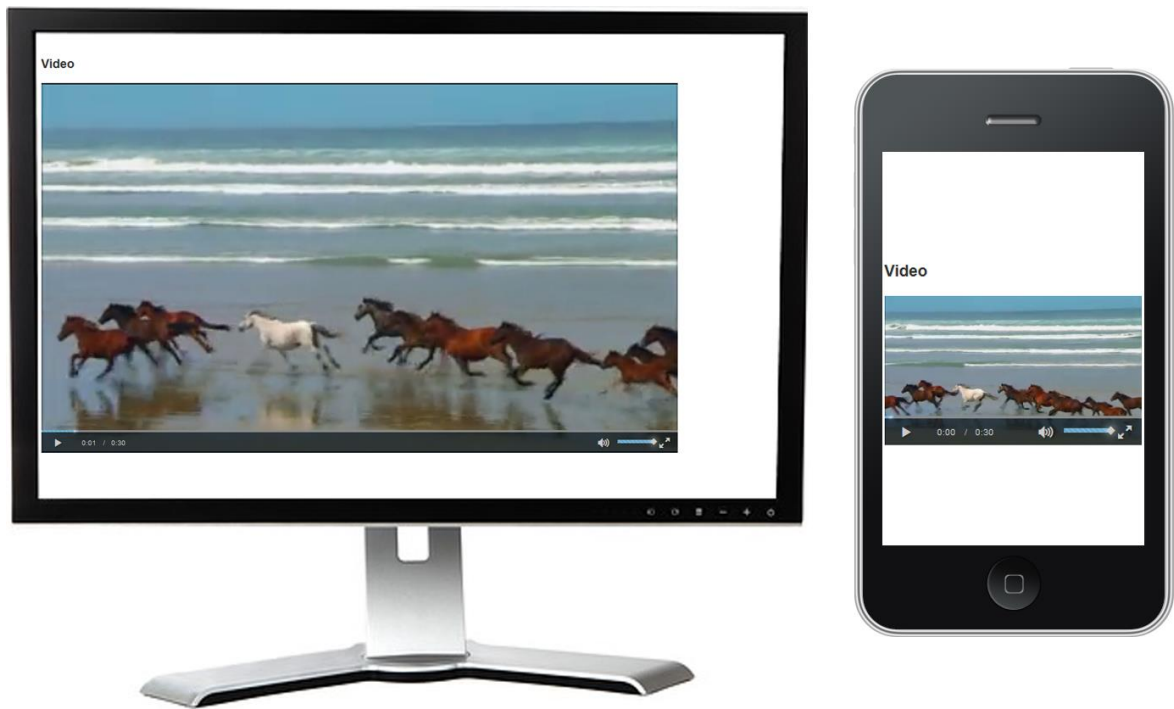
**Abbildung 3-10: Lightbox in Maximal-, und Minimalansicht**

### 3.2.5 Videos

Videos sollen sich stets flexibel ihre Größe, in Abhängigkeit vom Viewport, anpassen. Optimaler Weise werden sie in Abhängigkeit vom verwendeten Display, der verfügbaren Internetverbindung und des verfügbaren Viewports geladen werden. Ebenfalls ist eine Anpassung hinsichtlich der Bitrate denkbar<sup>1</sup>.

---

<sup>1</sup> Weitere Informationen: <http://www.fierceonlinevideo.com/story/adaptive-bit-rate-yellow-brick-road-or-fools-gold-hd-streaming/2011-01-28>



**Abbildung 3-11: Video in Maximal-, und Minimalansicht**

### 3.2.6 Tabellen

Tabellen nehmen stets die komplette zur Verfügung stehende Weite ein und skalieren sich bei deren Änderung über ihre Spaltenbreite.

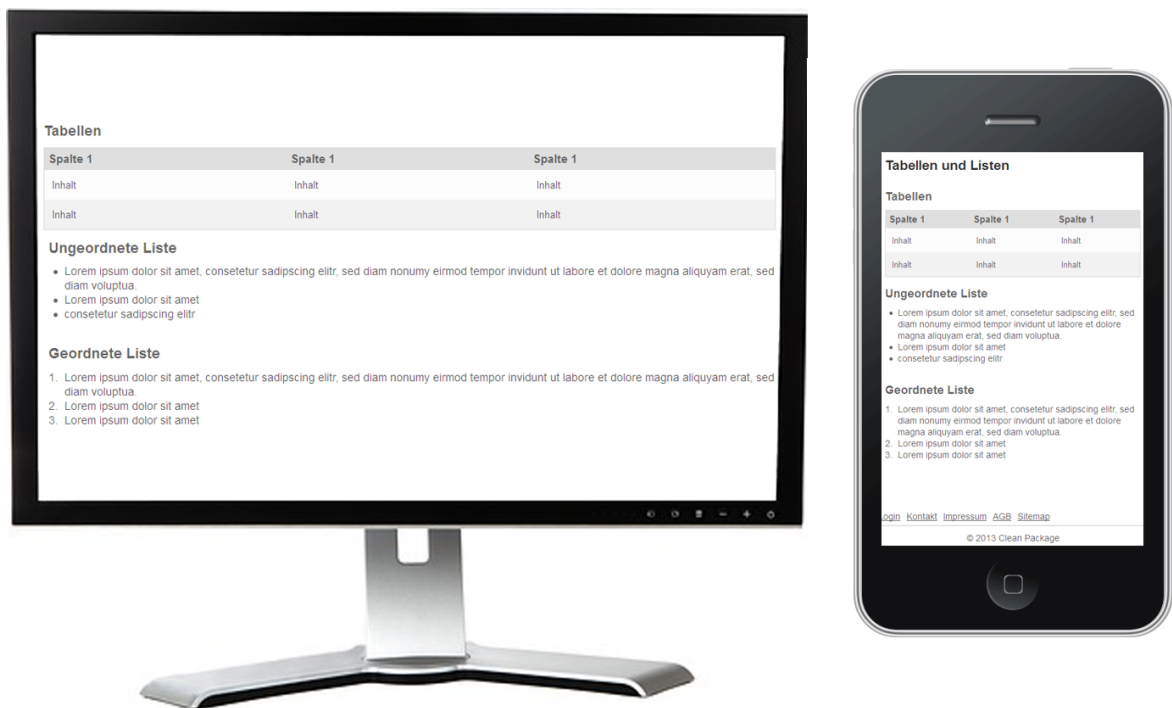


Abbildung 3-12: Tabelle und Listen in Maximal-, und Minimalansicht

### 3.2.7 Listen

Listen nehmen stets die komplette zur Verfügung stehende Weite ein und skalieren sich bei deren Änderung durch Textumbrüche.

### 3.2.8 Formulare

Formularelemente werden derzeit zweispaltig dargestellt. Bei Verringerung der Fensterbreite sollen sich Textfelder, Bilder (Captcha<sup>1</sup>), Auswahlfelder und mehrzeilige Texte horizontal und vertikal entsprechend mitskalieren. Nach Unterschreitung des zweiten BP wird das Formular einspaltig dargestellt.

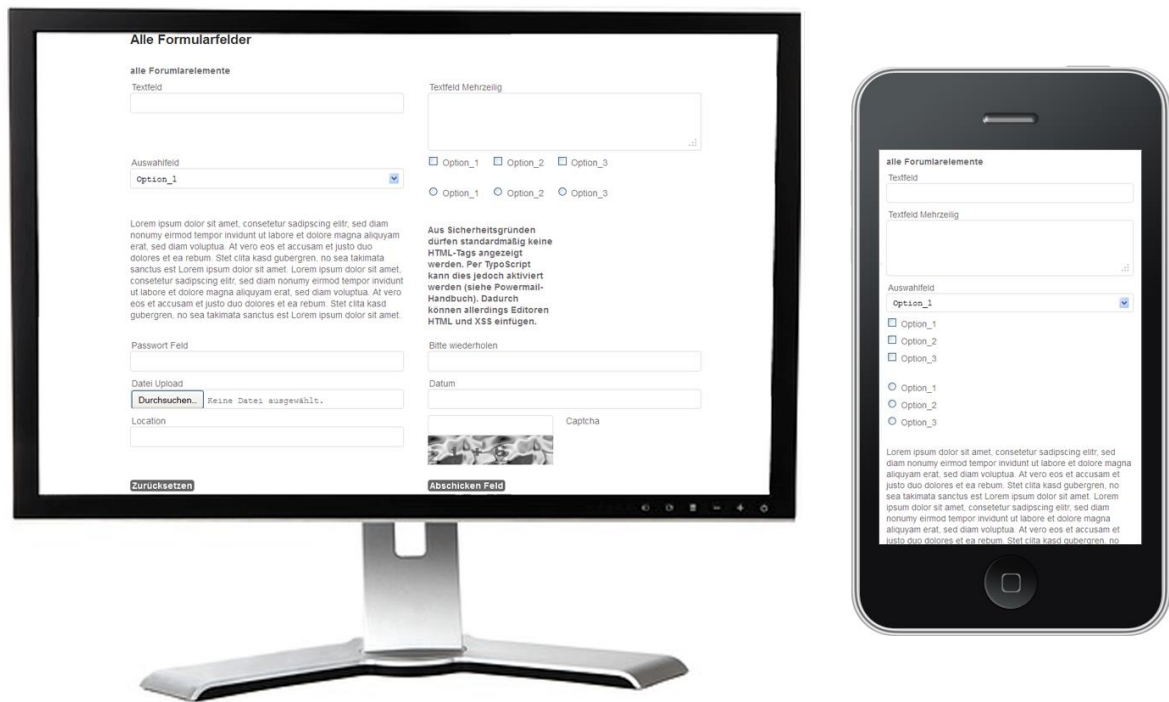


Abbildung 3-13: Formular in Maximal-, und Minimalansicht

<sup>1</sup> <http://www.heise.de/thema/Captcha>

## 4 Technische Konzeption

Den im letzten Kapitel festgelegten Anforderungen werden im Folgenden verschiedene Umsetzungsmöglichkeiten gegenüber gestellt. Basierend darauf wird vereinbart, wie die Umsetzung anhand der einzelnen Inhaltselemente erfolgt.

### 4.1 Frameworks

Im Internet kursieren einige Frameworks zu Responsive Webdesign. Für deren Nutzung spricht, dass man viele vorgefertigte Elemente in das eigene Template übernehmen kann. Dadurch verringert sich der Entwicklungsaufwand bei größeren Projekten erheblich.

In die engere Auswahl kommen hierfür das von Twitter entwickelte Bootstrap<sup>1</sup> sowie Foundation<sup>2</sup> der Firma Zurb. Beide Frameworks nutzen CSS Präprozessoren. Diese besitzen eine eigene Syntax, welche anschließend in CSS übersetzt wird. Der Vorteil liegt darin das man Styles global anpassen kann und er Code minimiert und übersichtlicher wird<sup>3</sup>. Die in den Frameworks enthaltenen Module, Gridsysteme und von vielen Entwicklern erprobten Erfolgsmethoden garantieren zusätzlich eine hohe Qualität der genutzten Elemente.

Der Nachteil, bei der Verwendung von Frameworks, besteht bekannter Maßen darin, zusätzlich ungenutzte Funktionalitäten zu implementieren, welche sich wiederum negativ auf Performance und Speicherbedarf auswirken. Bootstrap und Foundation bieten jedoch den entscheidenden Vorteil dass man bereits vor dem Download alle Komponenten initial festlegen und konfigurieren kann. Dadurch wird die Menge an zu implementierenden Code minimal gehalten.

In Absprache mit dem Projektfeld wurde sich letztendlich, aufgrund der größeren Community und der damit verbundenen Zukunftsfähigkeit sowie dem größeren Funktionsumfangs, auf Bootstrap festgelegt<sup>4</sup>.

---

<sup>1</sup> weitere Informationen: <http://getbootstrap.com/>

<sup>2</sup> weitere Informationen: <http://foundation.zurb.com/>

<sup>3</sup> weitere Informationen: <http://madd-agency.de/blog/css-preprozessoren-warum-sie-sinnvoll-sind>

<sup>4</sup> weitere Informationen: <https://medium.com/frontend-and-beyond/8b3812c7007c>

## 4.2 Layout

Die Breakpoints werden durch entsprechende Media Queries per CSS realisiert. Bei Verwendung von Bootstrap geschieht dies initial bei der Konfiguration. Die entsprechenden CSS Klassen müssen dann im Template vergeben, oder alternativ, bereits bestehende Klassen in den Media Queries angepasst werden.

### 4.2.1 Spalten

Hierfür müssen ebenfalls entsprechende CSS Anweisungen per Media Querie festgelegt werden, um das gewünschte Spaltenverhalten zu erreichen. Bootstrap nutzt hierfür ein Gridsystem<sup>5</sup>. Für dessen Verwendung müssen die entsprechenden CSS Klassen in das HTML-Template übernommen bzw. gerendert<sup>6</sup> werden.

### 4.2.2 Navigation

Bootstrap stellt bereits alle geforderten Navigationsarten responsiv zur Verfügung. Um diese zu nutzen, müssen die HTML-Navigationsstruktur sowie die entsprechenden CSS-Klassen angepasst werden. Alternativ können selbst festgelegte CSS Regeln innerhalb von Media Queries gesetzt werden. Ebenso existieren einige freie, auf JavaScript basierende Lösungen wie Beispielsweise Responsive Nav<sup>7</sup> oder Stashy Toggle<sup>8</sup>, im Internet. Für die geforderten Ausrichtungen, das Umbruchverhalten der Menüpunkte sowie Positionierung sind weitere CSS Anpassungen nötig.

## 4.3 TYPO3 Inhaltselemente

Einige Inhaltselemente, wie „Text mit Bild“ und „Video“, werden nicht durch das Framework abgedeckt und erfordern spezifische Lösungswege.

### 4.3.1 Typografie

Durch die Nutzung der Frameworks verhält sich der Text flexibel und die Schriftarten werden je nach Displaygröße skaliert. Bei einer händischen Lösung via CSS-Regeln

---

<sup>5</sup> zu Gridsystemen siehe: <http://maddesigns.de/responsive-grids-sass-1883.html>

<sup>6</sup> zu TYPO3 Template Rendering siehe:  
<http://docs.typo3.org/neos/TYPO3NeosDocumentation/IntegratorGuide/PageRendering.html>

<sup>7</sup> weitere Informationen: <http://responsive-nav.com/>

<sup>8</sup> weitere Informationen: <http://stashy.azurewebsites.net/toggle>

sollten ausschließlich skalierbare Einheit wie „em“ zum Einsatz kommen. Die Skalierung selbst sollte dann über einen globalen Prozentsatz im Body erfolgen<sup>9</sup>.

Überschriften hingegen werden initial beim Konfigurieren von Bootstrap festgelegt. Bei Verzicht auf das Framework müssen die Größenangaben per Media Query im CSS eingetragen werden.

### 4.3.2 Texte mit Bildern und News

Da diese TYPO3-Inhaltselemente, bestehend aus Text und Bild, nicht von Haus aus durch die Frameworks abgedeckt sind, bedarf es weiterer Anpassungen im Stylesheet. Für umfangreichere Projekte sollte man die entsprechenden Framework-Klassen für Bilder und Texte rendern und deren Positionierung per CSS definieren. Für das Cleanpackage ist dies jedoch nicht nötig. Gleiches gilt für das Inhaltselement „News“.

### 4.3.3 Bilder

Um die von Viewportabhängige Skalierung der Bilder zwischen den Breakpoints zu erreichen, bedarf es zusätzlicher CSS Regeln. Für die geforderte adaptive Bildanzeige muss zusätzlich das kostenlose JavaScript Plugin „responsive-images.js“<sup>10</sup> eingebunden werden. Im Vergleich zu anderen Plugins nutzt dieses, die seit der letzten TYPO3 Version 6.2 renderbaren, HTML5 data-scr-Attribute<sup>11</sup>.

Für Slider soll das ebenfalls kostenlose Plugin magnificPopup<sup>12</sup> genutzt werden. Dieses basiert auf CSS3-Transitions<sup>13</sup> und verzichtet auf Performance-lastige JavaScript-Animationen. Ein weiteres Argument ist die Unterstützung von Retina-Displays.

---

<sup>9</sup> Praxiswissen Responsive Webdesign S.112 [TiKa2013]

<sup>10</sup> Weitere Informationen: <http://davidwalsh.name/responsive-design>

<sup>11</sup> Weitere Informationen: <http://www.drweb.de/magazin/css-und-data-dashes/>

<sup>12</sup> Weitere Informationen: <http://dimsemenov.com/plugins/magnific-popup/>

<sup>13</sup> Weitere Informationen: <http://t3n.de/news/magnific-popup-responsive-475592/>



## 5 Integration

Es wird davon ausgegangen, dass das Cleanpackage in der TYPO3 Version 6.2.0 aus dem Confluence<sup>1</sup> ordnungsgemäß installiert und konfiguriert vorliegt.

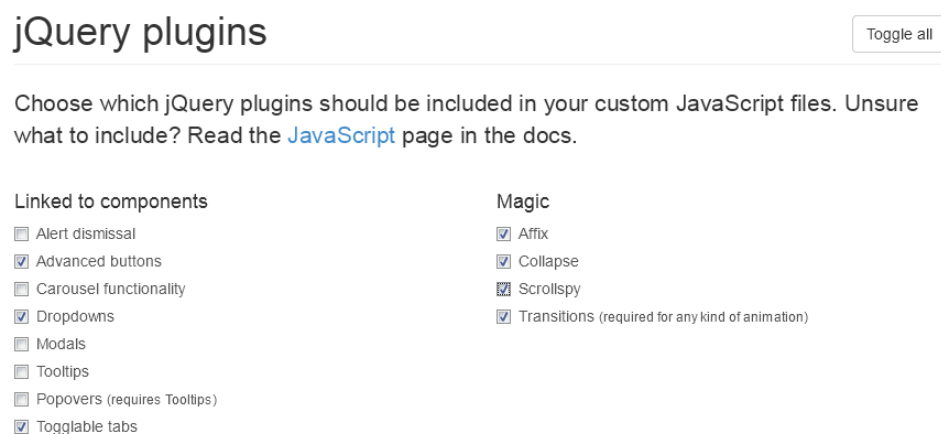
Nachfolgend werden nun die nötigen Schritte zur Integration des Bootstrap Frameworks und die weiteren Konfigurationen zur Umsetzung der technischen Konzeption dokumentiert.

### 5.1 Integration von Bootstrap

Um Bootstrap für die TYPO3 Inhaltselemente nutzen zu können, sind mehrere Schritte nötig.

#### 5.1.1 Konfiguration und Auswahl der Komponenten

Über das Konfigurations- und Downloadportal<sup>5</sup> von Bootstrap kann man selbst entscheiden, welche Komponenten man benötigt und welche überflüssig sind. Für die geforderten Anpassungen müssen die in Abbildung 5-1 und 5-2 ersichtlichen Komponenten ausgewählt sein.



jQuery plugins	Toggle all
Choose which jQuery plugins should be included in your custom JavaScript files. Unsure what to include? Read the <a href="#">JavaScript</a> page in the docs.	
<b>Linked to components</b>	<b>Magic</b>
<input type="checkbox"/> Alert dismissal	<input checked="" type="checkbox"/> Affix
<input checked="" type="checkbox"/> Advanced buttons	<input checked="" type="checkbox"/> Collapse
<input type="checkbox"/> Carousel functionality	<input checked="" type="checkbox"/> Scrollspy
<input checked="" type="checkbox"/> Dropdowns	<input checked="" type="checkbox"/> Transitions (required for any kind of animation)
<input type="checkbox"/> Modals	
<input type="checkbox"/> Tooltips	
<input type="checkbox"/> Popovers (requires Tooltips)	
<input checked="" type="checkbox"/> Toggleable tabs	

**Abbildung 5-1: Bootstrap jQuery Komponenten**

---

<sup>1</sup> Firmeninterne Wiki-Software der Firma Atlassian: <https://www.atlassian.com/de/software/confluence>

## Less files

[Toggle all](#)

Choose which Less files to compile into your custom build of Bootstrap. Not sure which files to use? Read through the [CSS](#) and [Components](#) pages in the docs.

### Common CSS

- ☒ Print media styles
- ☒ Typography
- ☒ Code
- ☒ Grid system
- ☒ Tables
- ☒ Forms
- ☒ Buttons

### Components

- ☐ Glyphicons
- ☒ Button groups
- ☒ Input groups
- ☒ Navs
- ☒ Navbar
- ☒ Breadcrumbs
- ☐ Pagination
- ☐ Pager
- ☒ Labels
- ☐ Badges
- ☐ Jumbotron
- ☒ Thumbnails
- ☒ Alerts
- ☐ Progress bars
- ☒ Media items
- ☒ List groups
- ☐ Panels
- ☐ Wells
- ☒ Close icon

### JavaScript components

- ☒ Component animations (for JS)
- ☒ Dropdowns
- ☐ Tooltips
- ☐ Popovers
- ☐ Modals
- ☐ Carousel

### Utilities

- ☒ Basic utilities
- ☒ Responsive utilities

**Abbildung 5-2: Bootstrap LESS Komponenten**

Des Weiteren können viele Styling-Regeln zu Typografie, Tabellen, Formularen, Listen und den Navigationselementen aufgestellt werden.

#### **5.1.1.1 Media Queries breakpoints**

An dieser Stelle kann man die entsprechende Weite von bis zu vier Media Queries festlegen. Da drei verschiedene Ansichten (Maximal-, Zwischen- und Minimalansicht) vereinbart wurden, werden für das Cleanpackage zunächst lediglich zwei Breakpoints benötigt.

#### **5.1.1.2 Gridsystem**

Hier wird die Anzahl der verwendeten Grids, deren Breite sowie der Breakpoint, welcher das Umschalten der Hauptnavigation in ein Pulldown-Menü definiert, festgelegt.

### **5.1.2 Implementierung in TYPO3**

Sind alle Konfigurationen getätigt, so kann man sich über den „Compile and Download“-Button sein angepasstes Framework als .zip-Datei herunterladen. Diese beinhaltet alle

nötigen JavaScript und CSS-Dateien jeweils in strukturierter sowie minifizierter<sup>2</sup> Version. Diese müssen nun auf den TYPO3 Webserver in folgende Verzeichnisstruktur hochgeladen werden.

CSS Dateien	CleanPackage\htdocs\fileadmin\layout\css\bootstrap\
JS Dateien	CleanPackage\htdocs\fileadmin\layout\js\bootstrap\

**Tabelle 5-1: Bootstrap Verzeichnisse**

Um diese nun im Template zu integrieren müssen folgende Dateien um einige Codezeilen ergänzt werden.

#### **5.1.2.1 Einbinden der CSS-Dateien**

Datei	CleanPackage\htdocs\fileadmin\layout\typoscript\setup\pages\page_css.ts
Anpassung	<pre> page.includeCSS {   bootstrap = {\$mms.site.cssPath}/bootstrap/bootstrap.css   bootstrapTheme= {\$mms.site.cssPath}/bootstrap/bootstrap-theme.css }</pre>

**Tabelle 5-2: Bootstrap CSS Implementation**

<sup>2</sup> Weitere Informationen: <http://blog.n-image.de/2012/08/javascript-und-css-dateien-komprimiert-ausliefern/>

### 5.1.2.2 Einbinden der JavaScript-Dateien

Datei	CleanPackage\htdocs\fileadmin\layout\typoscript\setup\pages\page_js.ts
Anpassung	<pre> page.includeJSFooter { bootstrap = {\$mms.site.jsPath}/bootstrap/bootstrap.js bootstrap.compressed = 1 } </pre>

Tabelle 5-3: Bootstrap JavaScript Implementation

### 5.1.3 Templateanpassungen

Nun sollte es möglich sein auf die Bootstrap-eigenen Komponenten, welche unter 5.1 konfiguriert wurden, zuzugreifen. Dafür ist es nötig die entsprechenden CSS-Klassen im TYPO3-HTML Template einzufügen. Zusätzlich muss für die Nutzung der Bootstrapnavigation die HTML Struktur analog zu Abbildung 5-3 angepasst werden

```

<div class="navbar navbar-default" role="navigation">
  <div class="container-fluid">
    <div class="navbar-collapse collapse">
      <ul class="nav navbar-nav">
        <li class="active">
        <li>
        <li>
        <li class="dropdown">
      </ul>
      <ul class="nav navbar-nav navbar-right">
    </div>
  </div>
</div>

```

Abbildung 5-3: Bootstrap HTML Navigationsstruktur

## 5.2 Integration magnific Popup

Die responsive Lightbox „magnific Popup“ besteht ebenfalls aus einer JavaScript Bibliothek sowie einem Stylesheet. Diese müssen zunächst in folgendes Verzeichnis des Webservers geladen werden.

Datei	Zielverzeichnis
jquery.magnificPopup.css	CleanPackage\htdocs\fileadmin\layout\css\jquery_plugins\
jquery.magnific-popup.min.js	CleanPackage\htdocs\fileadmin\layout\js\jquery\plugins\

Tabelle 5-4: magnific Popup Verzeichnisse

Um das Script zu Verwenden sind folgende Dateiergänzungen nötig:

Datei	CleanPackage\htdocs\fileadmin\layout\typoscript\setup\pages\page_css.ts
Anpassung	<pre> page.includeCSS { magnificPopup = {\$mms.site.cssPath}/jquery_plugins/jquery.magnificPopup.css } </pre>

Tabelle 5-5: magnific Popup CSS Implementation

Datei	CleanPackage\htdocs\fileadmin\layout\typoscript\setup\pages\page_js.ts
Anpassung	<pre> page.includeJSFooter { magnificPopup = {\$mms.site.jsPath}/jquery/plugins/jquery.magnific-popup.min.js magnificPopup.compressed = 1 } </pre>

Tabelle 5-6: magnific Popup JavaScript Implementation

Nun sollte die Lightbox verfügbar sein. Man muss nun das Verhalten eines Bild im TYPO3 Backend auf „Bei Klick vergrößern“ einstellen um diese zu aktivieren.



## 6 Fazit

Zukünftig führt kein Weg mehr an responsiven Webseiten vorbei. Die technische Entwicklung von internetfähigen Geräten geht rasant voran und dringt in alle Bereiche des Alltags. Smartwatches, Kühlschränke mit integrierten Displays und die Verwendung von OLED Displays auf Kleidung sind längst keine Zukunftsmusik mehr.

Aktuell sind Lösungen für bestehende Content Management Systeme oftmals nur Stückwerk hin zu wirklich responsiven Seiten. Zu der, in dieser Arbeit verwendeten, Projektvorlage konnte parallel ein Prototyp entwickelt werden welcher viele aufgeführten Ansätze bereits erfüllt und in den nächsten Wochen fertig gestellt wird.

Um das Thema weiter voran zu bringen müssen die Entwickler Standards schaffen durch welche es einfacher wird, den Nutzern von mobilen Geräten eine optimale User-Experience zu bieten.





## Literatur und Quellenverzeichnis

- [StHay2011] Hay, Stephan:  
<https://twitter.com/stephenhay/status/23350345962889216>,  
verfügbar am 30.04.2014, 15:00Uhr
- [EtMa2010] Marcotte, Ethan:  
<http://alistapart.com/article/responsive-web-design>, verfügbar am  
30.04.2014, 15:00Uhr
- [TiKa2013] Kadlec, Tim: Praxiswissen Responsive Webdesign,  
Köln, O'Reilly Verlag, 2013
- [ChZil2013] Zillgens, Christoph: Responsive Webdesign,  
München, Carl Hanser Verlag, 2013

## **Selbstständigkeitserklärung**

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 30. April 2014

Jens Wittig